Minutes of the Online phone meeting held on Oct 25

General Remarks

The collective task for the online group is to deliver an online system for three MINOS detectors:

1. The Calibration Detector at CERN (CalDet)
   Installation scheduled to begin April, 2001. Online system installation ~ May 2001.
2. The Far Detector at Soudan (FD)
   Beneficial Occupancy scheduled for May 1, 2001. Online system installation begins ~ June, 2001.
3. The Near Detector at Fermilab (ND)
   Installation begins Aug/Sep 2002. Beam ON Oct, 2003.

The time scale for delivery of a basic online system is defined by the CalDet and FD to be May, 2001. Commissioning time at the CalDet will be short so the key will be to test, test, test. Any further delays in beneficial occupancy would provide additional (valuable) commissioning time at the CalDet but, currently, would not relax the delivery milestone for a basic system. We should use these phone meetings in addition to the collaboration meeting sessions to discuss the issues of delivering the MINOS online system, in particular the issues of integration.

Future Meetings

Regular phone meetings should be arranged for discussion of common issues. These should be followed up with a working group meeting at the Stanford collaboration meeting in December. An integration workshop is suggested for January at RAL (coordinated with the proposed installation workshop at RAL if possible).

Basic Online Components

It has been agreed that the CalDet only requires a very basic, minimal online system in order to install and operate. However, since the FD requires more components of the online very shortly after CalDet commissioning we should try to deliver as much as possible to the CalDet. It provides a very useful commissioning tool for the FD. The components of the online system to be considered are listed below. Items in red are the basic CalDet system. Items in blue extend this but are non-critical to CalDet operation, at least over the first few months. We must provide items in red and should task ourselves to provide the items in blue if possible (note that there are correlation's between the blue components).

- **Data Acquisition & Triggering – DAQ group**
  - Basic data taking on a 2 or more crate system
    - Normal Data
    - Pedestals
    - Charge Injection
    - Light Injection
  - Basic Singles Monitoring
  - Basic run control (start, stop)
  - Triggering
    - Detector diagnostics
    - Cosmic rays }
    - Plane trigger for CalDet } when detector installed
    - Special CalDet triggers (Beam counters, cosmics counters) }
- **Timing System, including NTP – Alfons, Colin, Giles**
  - Fast timing services to FE/DAQ
  - NTP services for 2 or more crate DAQ system
  - NTP services to other online systems

- **Detector Control System (DCS) – DCS group**
  - HV control and monitoring
  - Monitoring data to DCS ROOT file online
- **Light Injection System – Lisa Falk-Harris**
  - Basic L.I. sequencing control – could be semi-manual
  - Real time analysis in trigger farm
  - All other analysis is offline
- **Online Database – Pete Border**
  - Definition of *online* database tables
  - Online database read/write access; down(up) loads to(from) the master
  - Online DBU (Database Update) process for DAQ, DCS -> DB information flow
  - Database read/write access for quasi online

  An online database is not strictly a necessity but should be provided if available. Database updating can all be performed offline under the online system design and system configuration is local
- **Electronic Logbook**
  - Local electronic logbook
- **Dispatcher and (ROOT) Raw Data Services - Robert Hatcher, Sue Kasahara**
  - Ability to create a raw data ROOT file
  - Local dispatcher services. Remote dispatching not required.

  The dispatcher is not a necessity for the CalDet but should be provided if possible
- **Online Monitoring – David Petyt**.
  - Framework
  - Interface to DAQ (+DCS?) data
  - Monitoring of basic quantities for detector checkout and CalDet operation

  Online Monitoring is not a necessity at the CalDet but should be provided if possible. Monitoring can be performed with a sufficiently fast turnaround offline at the test beam.
- **Online Event Display – Brett Viren plus physicist programmers (see below)**
  - Framework
  - Basic online event displays

  A basic event display exists at UCL which could be used offline if the MINOS display was not available. The MINOS display should be provided if at all possible however.
- **Data Handling – Storage & Distribution – Liz Buckley Geer**
  - The MINOS distribution system for getting the data to persistent store at Fermilab.

  This is not essential at all. Manual FTP of run files or posting of tapes is the minimum.

A discussion ensued on a number of these online components until the phone meeting timed out. Discussions will continue in future meetings.

Data Acquisition

The basic data acquisition system required for CalDet and FD commissioning covers essentially all the basic readout and control functionality of the system. Code development is under way (and on schedule) on the vertical slice system being assembled at RAL. This will test the readout, FE control, FE calibration (pedestals, charge injection), data routing and farm software. The vertical slice is a single crate system containing one (two later) VARC reading out 3 PMTs from a calibration detector style MUX box. The UCL mini module (128 channel steel/scintillator) with a prototype pulser box is at RAL and will be used to supply controlled and realistic signals to the system. The pulser box will allow light injection control and analysis software to be tested. The schedule for completing the vertical slice tests is the end of the year. We anticipate extending this to a two crate system when the vertical slice test is complete in order that two crate readout can be tested on real hardware – an important step. This system will be a valuable MINOS resource and offers a useful platform for wider online integration tests.

In addition to the DAQ system software, interfaces and integration with other systems are vital.

- DAQ <-> DCS
  The concepts of this interaction have long been agreed (see previous DAQ and DCS reviews). The two systems will be kept as much at arms length as possible. The intention is to have one interface only between the two systems (between the high level run control in the DAQ and the DCS PC) through which all information flows. Communication will be through sockets (messages and small volumes of monitoring data) and files (for large data volumes, which may yet prove unnecessary). Jeff McDonald has demonstrated socket communications with a number of wrappers and no problems are anticipated. The DAQ group has discussed the DCS interaction as part of the run control design at one of their group meetings. Mark Thomson, who will be providing the high level run control for the DAQ (as distinct from low level) has begun work on MINOS and has taken part in these discussions. There is a consensus that the interaction with DCS can be restricted to a very small set of messages which we have started to define. However, to proceed further a meeting is needed with the DCS group to discuss the issues and agree on details. This meeting was agreed and a time will be set. One further issue in particular which we would like to discuss is what we should do with the temperature monitoring data that the DAQ gathers since this is DCS-type monitoring (both the VFBs and the ROPs have temperature sensors which will be read out by the DAQ).
  Note: This meeting subsequently took place on Oct 30.
- DAQ <-> Flasher
  The DAQ / Flasher interface was recently discussed in a meeting between the DAQ group and the light injection group (Lisa and Philip). Some details of the interface may iterate over the coming weeks but there is now a firm understanding.
- DAQ <-> Dispatcher
  The interface between the DAQ and the dispatcher needs to be fleshed out in more detail. Run control requires status monitoring of the dispatcher and the dispatcher requires run status information. The ROOT file format needs to be concluded.
- DAQ <-> Database
  The DBU process is required for writing DAQ tables to the database. The database tables required by the DB need to be defined in more detail
- DAQ <-> Calibration Detector
  The one remaining major issue is whether any non-standard (i.e. non-MINOS) electronics are required for special CalDet counters (e.g. TOF). This is under study.

<u>Timing System and NTP services</u>

These will be discussed at a future meeting when Giles Barr and Alfons Weber are present. NTP services to the online system in general (i.e. beyond the DAQ system) should be discussed.

<u>Detector Control System (DCS)</u>

The Calibration Detector requires, as a minimum, HV control and monitoring. This can be achieved without the DCS system but the DCS group intend to supply DCS to the CalDet. The interface between DAQ and DCS was discussed briefly (see the DAQ section above) and will be the subject of a phone meeting next week. There was some discussion of the (ROOT) output files from the DCS and the interface to the dispatcher and data archival processes (the complication being that DCS runs on windows machines). Jeff McDonald said that the provisional plan was to write the DCS ROOT files to a disk on a Linux PC from where they could be accessed by the relevant online processes. It was not decided what PC this would be but it would need to run a DCS process to receive the data. It was felt that there would be advantages to having these files written to the same disk as the DAQ event files although the implications of mixing DAQ and DCS like this were not immediately attractive to the DAQ group. RAID disks are a possibility. This issue needs further thought and will be revisited.

The DCS output files will be dominated totally by beam data at the Near Detector where it is estimated there will be 120 GB/year of data (~ 3 Kwords/spill). Jon Urheim subsequently distributed an estimate of the beam data from Al Erwin which is included here in Appendix 1. The file sizes at the Far Detector and the Calibration Detector will be very small but these files need to be defined. The CalDet group have said they have no need for beam data to be recorded by the DCS though this should be confirmed by the DCS group. Any beam settings will be recorded in run comments (which will be written to the database via the DAQ – Database interface).

Light Injection

The DAQ / Flasher interface was recently discussed in a meeting between the DAQ group and the light injection group (Lisa and Philip). This discussion was written up and circulated. The potential (singles) rate of light injection data into the DAQ system remains a concern, particularly at the Near Detector where cross talk on the M64s can greatly increase the data volume. The capability for one complete detector calibration every hour is required by the scintillator/PMT groups. Sympathetic sequencing of the LED flashing is very helpful and Philip Harris is constructing a scheme which he will circulate. The real time software required for processing the flasher data in the trigger farm is under development by Lisa and will be ready for integration tests with the DAQ on the vertical slice at RAL by December. This processing will produce one summary record from each time frame containing flasher data, the record containing a summary for each of ~ 600 channels. Lisa has a preliminary definition of the flasher summary data and an estimate of the summary data volume which she promised to circulate after the meeting (this is attached as Appendix 2). The volume of data was a surprise to a number of people but should be handle-able. Processing of the summaries will be performed offline (the possibility of some quasi online processing remains) with the resultant calibration data going to the database.

Pete Border was concerned at the prospect of the database having to handle fresh detector calibrations on an hourly basis. The problem is not the data volume per se (Oracle can store semi infinite amounts) but the prospect of hourly distribution around the collaboration. Pete will continue this discussion offline.

Control software for the LI system (for the LI PC) is less well developed than the farm processing software but will be made available on the required time scale by Lisa. This can be tested on the prototype pulser box in the RAL vertical slice.

Phil and Lisa confirmed that the light injection system has no requirement on the DCS. Control of the Light Injection mode will be executed directly from Run Control. A local interface (on the L.I. PC) will be used for expert configuration. Processing output, including LED information etc, will be via the DAQ data stream. Calibration constant updating of the database will occur offline.

Online Event Display

Jon Urheim reported that the MINOS event display framework is to be developed by Brett Viren. The plan is still to have a single event display for MINOS which serves both offline and online needs. Further physicist programmer effort is required for the development of displays within this framework and this is actively being sought. This will be a crucial issue for the Stanford meeting in December after which a full team should be in place.

Data Handling

Given that the data volumes from the Calibration Detector are relatively small, Liz will strive to provide the data storage system being developed at Fermilab for MINOS to the CalDet. This will provide archival of data to the Fermilab mass storage system over the network and will be a good test of the system *in the field*. Calibration Detector data will immediately become a formal component of the MINOS data set and be available to the collaboration via the MINOS distribution system. Data buffering capability at the CalDet is large compared with the data rates so network performance should not be an issue.

Raw Data Services

Robert is designing the raw data objects for the ROOT file based on the draft DAQ data formats and the recent light injection summary format. Drafts of this have been presented and discussed at previous meetings. Robert noted that provision for floats as well as integers is required for which some further thought needs to be applied. A definition of the DCS output records is needed to do the same for the DCS ROOT file.

Online Database

Pete has presented his plans for database provision to the Far Detector at his reviews and has distributed his plans and questions for provision at the Calibration Detector in recent weeks. These are attached in Appendix 3. One of the main issues for other systems is the definition of those tables that are required in the database for both the FD and the CalDet and an estimate of sizes. Sub system managers are urged to communicate with Pete on this to define, at least, what the full set of tables will be. CalDet issues can also be routed through Paul Miyagawa, the CalDet database liaison / manager.
Concerns on the impact of hourly detector calibrations on the database were expressed (see light injection above) and will be followed up with discussions after the meeting.

Electronic Logbook

This will be required for Soudan installation and should be provided for the Calibration Detector if possible. They can use paper at startup but an electronic logbook from day 1 will be advantageous.

Appendix 1 – Al Erwin Estimate of end-of-spill data size

Here is a preliminary list of end-of-spill data that the DCS might read out from a socket in an Accelerator Division server. I put together this list by polling beam line physicists in MINOS. So far it looks like about 748 words, but I expect other candidates to trickle in.

1. Proton Count
2. Last 2 Beam Position Monitors (BPM's). 2 x's and 2 y's
3. An OTR (Optical Transition Radiation) monitor.
   There would be 2 of these. Each has a 20 X 20 CCD pixel read out. This implies 2 X 400 words.
4. Two thermocouple monitors (left & right) to detect beam wandering.
   (Couple of minutes to reach equilibrium)
5. One word for the low energy beam Budal monitor.
   Another word if we the targets for the higher energy beams.
6. 4 words for the horn currents.
7. 4 words for the horn timing.
8. 2 words for Hall probe and B-field pickup.
9. 1 word for the microphone.
10. Flow meter for water. How many?
11. Water temperature. How many?
12. Water level in tank.
13. Temperature in the target pile.
14. Hadronic Hose: 4 currents & 4 timings.
15. Cooling air temperature: IN and OUT
16. Decay pipe vacuum.
17. Beam loss monitors. How many?
18. Exhaust stack monitor for air activation.
19. Decay pipe temperature. How many?
20. Absorber temperatures. 8 Al pieces, 1 steel that are water cooled. 3-5 pieces of steel not water cooled.
21. Two muon beam monitoring chambers (96 words per chamber).
    Possibly a hadron chamber with another 96 words.

Albert Erwin

Appendix 2 – Light Injection summary output

Dear Onliners,

As promised during yesterday's phone meeting, I'm passing around a draft of the format for the light injection summary events. Please feel free to comment on it!

First a brief recap of what it's all about:

The light injection calibration system will run an analysis process on the trigger farm. This process will be passed all the light injection events in a time frame. It will calculate simple statistics of these events and produce a summary event that will be placed in the output data stream. Under normal circumstances, the original LI events will be discarded at the trigger farm and replaced by the summary event in the data stream. (There will, however, be a debug mode in which the summary analysis is shortcut and the original events kept.)

In normal operating mode, the summaries contain the mean and rms of the adc counts for each channel. There will also be a mode in which the summaries include timing statistics. The LI system will be used to measure the relative timing of the different channel. This mode will be used during startup and will probably be of little interest thereafter.

Summary format:

The format of the summaries will follow the general MINOS block structure. To the best of my present knowledge, will be as follows: One summary header followed by N channel summaries for the individual channels.

Summary header (not complete, and not necessarily in this order...):
- LED number
- No of channels in summary
- Timing info included in this summary (yes/no)
- [Probably a time stamp + a few spare fields]

Channel summary #0:
- Channel no (Crate no and the 13 address bits used in the data word)
- ADC sub-summary:
  - No of entries
  - Mean
  - RMS
- Timing sub-summary (when applicable):
  - No of entries
  - Mean
  - RMS

Channel summary #1:
- .....
- .....

Channel summary #N-1:
- .....

About the size of these summaries:

Allowing some safety margin at the moment, I estimate that the summaries will take up 12 bytes/channel + a header under normal conditions; when the timing info is required, the size of a summary would be about 20 bytes/channel + the header. This can probably be slimmed a bit; it's quite likely that the summary for one channel could be fitted into 10 and 16 bytes respectively. The worst case in the FD is 640 strips lit up by one LED; with the two sides read out and a cross-talk factor of 2 we end up with 2560 channels to read out. The size of a summary would then be of the order of 30 (50) Kbytes, if we allow for 12 (20) bytes/channel.

We don't seem quite to have settled on the rate at which the LI system will flash just yet. High-frequency flashing at long intervals means that the summaries can contain the averages of many individual flashes and so occur less often in the output data stream. There are, however, constraints on the frequency at which we can flash, put by the data throughput capacity at the crates. I think it's therefore safest to assume at this stage that a summary will be produced for essentially every time frame.

Lisa

Appendix 3 – CalDet Database Issues

Hi:
Here are my comments on database for the Calibration module:

Tables are likely to be very difficult at the Cal module: it's a test run, so people will be changing things and expecting completely new versions of data structures on overnight time-scales. Getting the database to keep up with new schemas is going to be challenging, to put it mildly. I expect we can keep things running if we set up a system designed to accommodate rapid change, but it's likely to be iffy for a while.

Anyway, here's a general proposal for DB at the test beam:

1. The DBMS will be MySQL. We don't need Oracle there, and we cant support it anyway.

2. "Lucky" Paul M will be the local contact guy. Pete B will be on the phone/Net.

3. Any data entering the DB from any source will have a sequence number which links to a table of DB additions- in the additions table are columns for

- data source (Calibration program? Hand-done tweaks?
  Survey people? alignment program?)
- procedure code (calibration scheme x? intuition?)
  this may be several columns, including links to
  key/value pairs for the calibration program,
  survey logs, and amount of wine fueling the
  intuitive leap.
- date/time added
- person adding data, required for hand tweaks.
- link to logbook entries
- comments, comments, comments (required for hand tweaks)

4. Data entering the DB will also have a validity date set, which is distinct from the dates in the add table above.

5. The tables will be constructed as requested by the test beam group. I expect they will want, at a minimum

- Run table, with start/stop dates, magnet set currents, run conditions, other header data, and, of course COMMENTS. Does trigger counter data go here?
- Calibration table, with parameterization data for each strip. We should not assume that the parameterization stays constant during the whole course of the test run- people may want to experiment with different formulas.
- Flasher data summaries. This is probably two tables, one for each flasher event and one for results.
- Tube monitoring summaries. Do people want this in the DB, or just the event stream?
- Beams data table, with magnet read currents, SWICS(?) from each spill.
- DCS HV and current set levels and alarms data. See the DCS tables proposed in the database docs.
- Algorithm configuration key/value pairs? Are these going to be used?
- Survey and alignment tables. Format is up to the calibration group.
- Plexus and Ugli tables. Do we need the full time-dependent Plexus system, or just the summary table? What format does the Ugli need from the database anyway?

6. Does the test run group want to use an electronic logbook or paper? There are advantages to either, especially considering that the paper log will work on Day 1, and the electronic one won't.

7. Revisions to tables may be requested; don't expect instant response. Technical note: adding new columns to existing tables in a database is easier than you might expect, so changing parameterizations is possible without major damage (we could just add some new columns for the new system, and leave the old ones in place. Data with the old system will have NULLs for the new columns; data with the new system will have NULLs in the old columns. Disk is very cheap nowadays- we can waste a lot before its a problem...).

8. DBU will have a real user interface; behind the curtain will be something else. DBU has never been meant to cascade or cache, so it's much easier than DBI to produce. We can use the TSQL-ODBC interface, or if there's a real crunch, code can be lifted from the NewRiders MySQL book which has everything we need to write directly into MySQL.

8.5 Users will enter data into the DB through a www-based system of PHP or perl pages. These systems are easy to make and will let us make sure we get the tables and sequence numbers done right. I do NOT want users poking around inside a relational database without some framework to make sure they keep all the links tidied up right.

9. DBI, OTOH, needs to be much more advanced. People will need the cascading facility, and will probably override official constants all the time. A flat-file ODBC reader is a must. I'm working on it...

Schema changes can be accommodated with the data in the additions table, I hope.

Maybe this has gone by already, and I missed it, but how does the candidate history know when a database value is overridden? What happens if a physicist decides he must override official constant X with his own X-prime in a flat file? Is this annotated in the history of all these candidates? Certainly the file used to override the cascader is gone after the job ends. Sorry if this has already been solved, I dont tune in to candidate discussions as well as I should.

I don't know if DBI needs caching for the test run- sounds like a crate level cache gets everything every time anyway..

Pete